

## Week 4 – Perbandingan RNN, LSTM, dan GRU Model

*Muhammad Naufal Firdaus\_1103220083*

### A. Pytorch Deep Learning Model

Arsitektur dari masing-masing model yang digunakan :

#### 1. RNNModel

- Terdiri dari 3 lapisan RNN bertingkat, ukuran hidden dimensi bertahap dikurangi.
- Diakhiri dengan dua lapisan fully connected.

#### 2. LSTMModel

- Terdiri dari 5 lapisan LSTM bertingkat.
- Dropout digunakan setelah LSTM dan sebelum fully connected layer.
- Hidden size lebih besar dari RNN (256 vs 128).

#### 3. GRUModel:

- Embedding layer
- 3-lapis GRU (tidak bidirectional)
- FC → Dropout → FC → Output
- Output berupa sigmoid probability

Adapun perbandingan matriks evaluasinya adalah sebagai berikut

Metriks Evaluasi	RNN	LSTM	GRU
Accuracy	0.7842	0.5000	<b>0.8851</b>
Precision	0.7698	0.5000	<b>0.8952</b>
Recall	<b>0.8108</b>	<b>1.0000</b>	0.8723
F1 Score	0.7898	0.6667	<b>0.8836</b>
AUC	0.8424	0.5306	<b>0.9543</b>

- **GRU Model** memiliki performa tertinggi di hamper semua metrics evaluasi, model ini lebih stabil dan efisien dalam menangani dependensi jangka panjang. Arsitektur dengan dropout dan dua FC layer mungkin membantu generalisasi dengan baik
- **LSTM Model** memiliki recall yang tinggi, artinya mendeteksi semua kasus positif namun precision yang rendah banyak false positive. Performa keseluruhan kurang baik mungkin karena overfitting atau stuck. Bisa juga karena *lr* atau epoch yang kurang sesuai
- **RNN Model**, dengan performa yang cukup, recall juga cukup tinggitalpi rentan terhadap vanishing gradient karena struktur dasarnya.

## B. Tensorflow Deep Learning Model

Arsitektur masing-masing model yang digunakan :

1. RNN Model :
  - a. **Embedding layer** untuk mengubah kata jadi vektor.
  - b. **5 lapisan SimpleRNN bertingkat** ( $128 \rightarrow 128 \rightarrow 64 \rightarrow 64 \rightarrow 32$ ), digunakan untuk menangkap pola urutan dalam teks.
  - c. Setelah tiap RNN: **Dropout** untuk mengurangi overfitting dan **BatchNormalization** untuk stabilisasi.
  - d. **2 Dense layer** (128 dan 64 unit) untuk transformasi fitur sebelum output.
  - e. Output layer **Dense(1, sigmoid)** untuk klasifikasi biner.
2. LSTM Model :
  - a. **Embedding layer** untuk mengubah input kata menjadi vektor.
  - b. **3 lapisan LSTM bertingkat** (128, 128, 64 unit) untuk menangkap urutan dan konteks panjang dalam teks.
  - c. Setiap LSTM diikuti oleh **Dropout** dan **BatchNormalization** untuk mencegah overfitting dan menstabilkan training.
  - d. **2 Dense layer** (128 dan 64 unit) untuk pemrosesan lanjutan fitur.
  - e. Output layer **Dense(1, sigmoid)** untuk klasifikasi biner (positif/negatif).
3. GRU Model :
  - a. **Embedding layer** untuk mengubah kata jadi vektor.
  - b. **3 lapisan GRU bertingkat** (128, 128, 64 unit) untuk menangkap konteks urutan kata.
  - c. Setiap GRU diikuti oleh **Dropout** dan **BatchNormalization** agar training stabil dan tidak overfitting.
  - d. Dilanjutkan dengan **2 Dense layer** (128 dan 64 unit) sebagai pemrosesan akhir.
  - e. Output berupa **Dense(1, sigmoid)** untuk klasifikasi biner (positif/negatif).

Perbandingan metrics evaluasi :

Metrik	RNN	LSTM	GRU
Accuracy	0.5010	0.7313	<b>0.8780</b>
Precision Avg	0.5121	0.8060	<b>0.8782</b>
Recall Avg	0.5010	0.7313	<b>0.8780</b>
F1 Score Avg	0.3518	0.7138	<b>0.8779</b>

- **GRU unggul di semua aspek:** Akurasi tinggi, precision & recall seimbang, dan performa yang stabil.
- **LSTM masih bagus**, tapi sedikit kalah dari GRU, mungkin karena arsitektur lebih kompleks dan sensitif terhadap overfitting.
- **RNN paling lemah**, karena tidak mampu menangkap dependensi panjang dalam teks dan cenderung overfitting atau underfitting pada data besar seperti IMDB.